Solving Classic AI Problems with Search Algorithms

1. Introduction

Artificial Intelligence (AI) search algorithms are fundamental techniques used to solve problems that involve finding a sequence of actions leading from an initial state to a goal state. This project aims to provide hands-on experience in implementing and analyzing classic search algorithms by applying them to a well-known AI problem.

In this project, we focus on a **Robot Pathfinding Problem in a Grid with Obstacles**. The robot must navigate from a starting position to a goal position while avoiding obstacles. Multiple search strategies are implemented and compared to understand their performance, efficiency, and suitability for different scenarios.

---

2. Problem Definition

2.1 Problem Description

The environment is represented as a two-dimensional grid. Each cell in the grid can be:

- Free space (0)
- Obstacle (1)
- Start position (S)
- Goal position (G)

The robot can move one step at a time in four directions: up, down, left, and right. Each move has a uniform cost of 1.

2.2 State Representation

A state is represented as the robot’s current position in the grid:

```
State = (x, y)
```

2.3 Initial State

The initial state is the cell marked with `S`.

2.4 Goal State

The goal state is the cell marked with `G`.

2.5 Actions

- Move Up
- Move Down
- Move Left
- Move Right

2.6 Path Cost

Each movement has a cost of 1. The total path cost is the number of steps taken from the start to the goal.

---

3. Grid Environment

The grid used in this project is shown below:

```
S 0 0 1 0
1 1 0 1 0
0 0 0 0 0
0 1 1 1 0
0 0 0 G 0
```

---

4. Implemented Search Algorithms

4.1 Breadth-First Search (BFS)

BFS explores the search space level by level. It guarantees finding the shortest path when all step costs are equal.

Characteristics:

- Complete and optimal
- High memory consumption

4.2 Depth-First Search (DFS)

DFS explores one branch of the search tree as deeply as possible before backtracking.

Characteristics:

- Low memory usage
- Not optimal
- May get trapped in deep paths

4.3 Uniform Cost Search (UCS)

UCS expands the node with the lowest path cost. It is optimal when all costs are non-negative.

Characteristics:

- Complete and optimal
- Higher time and space complexity

4.4 Iterative Deepening Search (IDS)

IDS combines the advantages of BFS and DFS by gradually increasing the depth limit.

Characteristics:

- Complete and optimal
- Lower memory usage than BFS
- Repeated computations

4.5 A* Search

A* is an informed search algorithm that uses a heuristic function to guide the search toward the goal.

Heuristic Function:
Manhattan Distance:

```
h(n) = |x1 - x2| + |y1 - y2|
```

This heuristic is admissible and consistent, ensuring optimal solutions.

Characteristics:

- Complete and optimal
- Fast and efficient

4.7 Genetic Algorithm

The Genetic Algorithm is an optimization technique inspired by natural selection.

Representation:

- Chromosome: A possible path
- Fitness Function: Inverse of path length

Operators:

- Selection
- Crossover
- Mutation

Characteristics:

- Good for optimization
- Does not guarantee optimal solutions

---

5. Performance Analysis

The algorithms were evaluated based on:

- Time complexity
- Space complexity
- Optimality
- Path cost

---

6. Comparison of Algorithms

Algorithm	Optimal	Time Complexity	Space Complexity
BFS	Yes	Medium	High
DFS	No	Fast	Low
UCS	Yes	Slow	High
IDS	Yes	Medium	Low
A*	Yes	Fast	Medium
Genetic	No	Medium	Medium

7. Results and Discussion

The experimental results show that **A*** provides the best balance between efficiency and optimality for the robot pathfinding problem. BFS and UCS guarantee optimal solutions but require more memory. DFS is faster but unreliable in finding optimal paths. The Genetic Algorithm is useful for optimization but does not always converge to the best solution.

8. Conclusion

In this project, multiple search algorithms were successfully implemented to solve a robot pathfinding problem. Each algorithm has its strengths and weaknesses. Informed search methods, especially A*, are the most suitable for this type of problem due to their efficiency and guaranteed optimal solutions.

This project demonstrates the importance of selecting the appropriate search strategy based on problem constraints and performance requirements.

9. References

- Russell, S. & Norvig, P. *Artificial Intelligence: A Modern Approach*
- Course Lecture Notes